A Practical Study on Shape Space and Its Occupancy in Negative Selection

Wanli Ma, Dat Tran, and Dharmendra Sharma, *Member and Senior Members, IEEE*

***Abstract*—The success of a negative selection algorithm depends on its detectors. A shape space, conceptually, is where selves, detectors, and antigens reside. These detectors are expected to fully cover the whole shape space. The better the coverage; the better the detection rate. However, this assumption brings a major challenge to negative selection experiments - the scalability problem, where the experiments cannot process real life datasets in a timely manner. On the other hand, with any real life dataset, due to arbitrary antibody/antigen representing formats, the shape space actually cannot be fully occupied. The unoccupied locations sometimes constitute a significant, or even overwhelm, portion in a shape space. In this paper, we first briefly review the theoretic model of the shape space and then study the impact of the unoccupied locations, under the term shape space occupancy. Based on the study outcomes, we suggest the heuristics for generating detectors. We demonstrate shape space occupancy, detector generation by antigen feedback mechanism, and negative selection experiments on 4 different datasets, which cover the data presentation formats in both strings and real number valued vectors.**

I. Introduction

Artificial Immune Systems (AIS) are inspired by the observation of the behaviours and the interaction of antibodies and antigens in a biological system [1, 2] and have attracted increasing interest from the research communities in the last 20 years [3-5].

Negative selection, which is a branch of AIS, mimics the way a human body detects and destroys harmful antigens. A human body constantly produces lymphocytes, with randomly mutated surface peptides, from bone marrow. All newly generated lymphocytes are sent to thymus to mature. The thymus has almost all types and shapes of self cells. During this period of maturing time, if a lymphocyte matches any cell in the thymus, it is destroyed. Only these which do not match any of the self cells in the thymus are sent to the body to match (or detect) antigens, which are invasion cells. The lymphocytes keep matching all the cells in the body. If a match happens, it means that a non-self cell (an antigen) is just detected. An alarm might be raised, and immune reactions may follow. The lymphocyte which matches the antigen may become a memory lymphocyte and stays in the body to quickly respond to the same antigen in the future. If for a period of time, a lymphocyte does not make any match, it will age and die. For the detailed explanation on how the immune system works, under the context of AIS, we refer the readers to [6, Chapter 2].

The terms used in AIS literature are yet to be standardized. In this paper, we use the terms antibody and detector interchangeably, and we view the data to be verified, i.e., to be matched by the antibodies, as a sequence of antigens (or nonselves). For the purpose of simplicity, we call all data items to be verified as antigens.

Negative selection has found wide applications, especially in abnormal detection [2, 7-10]. Ji and Dasgupta have a comprehensive survey paper [11] on the latest development of almost every aspect in this area.

Although being successful in some testing cases, negative selection experiments on real life datasets met a serious obstacle – scalability. In [12], Kim and Bentley reported the difficulties in generating useful detectors within an acceptable time window. Stibor et al [13-15] tested varieties of detector generation and affinity calculation methods and consistently pointed out that negative selection suffers from the scalability problem and is infeasible in real life applications. However, Balthrop et al [7] and González et al [9] believed that the problem is not with the negative selection algorithms themselves, but data representation formats and matching rules. In [21], Elberfeld and Johannes proposed an efficient algorithm for string-based negative selection with the theoretical up-bound for a single matching operation at $O(|D|^2 Lr)$, where $|D|$ is the number of detectors. There is yet an implementation of the proposed algorithm. However, from the theoretical result, we can see that the number of detectors plays a significant role in matching performance. In other words, *the number of detectors decides the scalability of the algorithm*.

We also believe that the scalability problem is not due to the negative selection algorithms themselves, but because of data representation formats and the matching rules. However, we do not think that changing data representation formats and/or matching rules will change the performance. A direct consequence of using any (arbitrary) data representation format is that a potion of the shape space, where selves, detectors, and antigens reside, cannot possibly be legitimately inhabited. The data, under the representation

Manuscript received January 31, 2010.

Wanli Ma is with the Faculty of Information Sciences and Engineering, University of Canberra, Australia (phone: +61-2-62012838; fax: +61.2.62015231; e-mail: Wanli.Ma@canberra.edu.au).

Dat Tran is with the Faculty of Information Sciences and Engineering, University of Canberra, Australia (e-mail: Dat.Tran@canberra.edu.au).

Dharmendra Sharma is with the Faculty of Information Sciences and Engineering, University of Canberra, Australia (e-mail: DharmendraSharma@Canberra.edu.au).

format, for some locations simply do not exist. Therefore, *the solution lies at finding a means to efficiently generate effective detectors*, instead of generating random detectors to cover the whole shape space.

Our solution stems from the study on the original shape space theory. The concept of shape space was first introduced by Perelson and Oster [16]. A shape space, conceptually, is where selves, antibodies, and antigens reside. Each of them can be represented as a point on the space. With some initial assumptions, one can establish the relationships among antigen detection rate, the number of required antibodies in the shape space, and the complexity of antibodies/antigens.

In this paper, we first briefly review the shape space theory developed by Perelson and Oster and then study the impact of shape space occupancy, especially, when a shape space is sparsely populated. Based on the outcomes of the study, we suggest the heuristics for generating detectors. Instead of trying to generate detectors to cover the whole shape space, we propose to generate effective detectors, which cover only the occupied portion of the shape space. The antigen feedback mechanism, initially proposed in [17], is a good example of the heuristics to generate effective detectors.

The rest of the paper is organised as follows. In Section II, we first briefly review the original shape space theory and then study the impact of shape space occupancy. By using KDD CUP 1999 dataset [18] as an example, Section III shows the shape space occupancy and also visualizes the landscape of the occupants in the shape space. In Section IV, we propose the heuristics for generating detectors and also discuss the antigen feedback mechanism. Section V reports the results of our experiments on 4 different datasets, with focusing on the performance of the antigen feedback mechanism on effective detectors and detection rates. We summarize the paper with future work in Section VI.

II. A Shape Space and Shape Space Occupancy

The concept of a shape space was first proposed and studied by Perelson and Oster [16] in 1979. It has been used by AIS research community as the foundation to analyse antibody generation algorithms and study antibody coverage.

A shape space Ω is a N dimensional sphere of radius R, where the antibodies (detectors), antigens, and selves reside. Each of them is represented as a point in Ω. The original authors suggested a region within the shape space of volume V and conducted all subsequent calculations based on it. The space outside of the volume V is completely ignored. In this paper, for brevity, we take the volume V as the whole shape space.

In a system, there are a fixed number of selves. The number of antibodies B can be estimated, as they are generated to cover Ω. Antigens can also be put into their corresponding locations in Ω. A metric is defined on Ω to calculate the distance between 2 points in the space. In the subsequent a few paragraphs, Euclidean metric is assumed; however, the conclusion is still valid if other metrics are used. If the distance between 2 points is smaller than ε, the 2 points are regarded the same. In other words, an antibody can detect any antigens which are within the ε distance from it. Finally, the dimensional parameter N can be regarded as the complexity of antibodies/antigens.

The volume of Ω is $c_N R^N$, where c_N is a constant, and an antibody can cover a volume $c_N \varepsilon^N$. Assume that the antibodies are distributed Poissonly in Ω, the average density of B number antibodies in Ω is:

$$\rho = \frac{B}{c_N R^N} \tag{1}$$

The probability that no antibody is within a volume $c_N \varepsilon^N$ is:

$$P_0 = e^{-\rho c_N \varepsilon^N} \tag{2}$$

and therefore, the probability for one or more antibodies within the volume $c_N \varepsilon^N$ is:

$$P = 1 - P_0 = 1 - e^{-\rho c_N \varepsilon^N} = 1 - e^{-B\hat{\varepsilon}^N}, \text{ where } \hat{\varepsilon} = \frac{\varepsilon}{R} \tag{3}$$

With one or more antibodies within the volume $c_N \varepsilon^N$, any antigen which falls into the volume will be detected. Therefore, P is also the probability for antigen detection, or detection rate.

In a shape space Ω, there also reside Q number of distinguishable selves, which are all known, unlike antigens. A hole of a N dimensional sphere of radius ε exists for every self. As the selves are all distinguishable, the holes of the selves do not overlap, if ε is small enough. Therefore, the total volume of the holes is $Qc_N \varepsilon^N$. The probability that a random antigen falls into the holes, thus cannot be detected, is $\frac{Qc_N \varepsilon^N}{c_N R^N} = Q\hat{\varepsilon}^N$. The original authors argued that $\hat{\varepsilon}$ must be small enough "*in order for the shape space not to consist mainly holes*". In the other words, if $\hat{\varepsilon}$ is small enough, it can counter Q, and thus not to void formula (3).

Although the formula is based on the assumption that the antibodies are distributed Poissonly within a R radius N dimensional sphere, it still is valid without this rigid assumption. We refer the readers to the original paper [16] for the discussion.

Furthermore, even if Formula (3) was derived on continuous space based on the calculation of the volume of N dimensional sphere, it is also valid for a discrete space where a point in the space is represented in a vector or a string format. For example, for a string format, if the alphabet set is Σ, and the length of a string is N, the shape space is then the collection of all possible N length strings Σ^N. The shape space can then be viewed as a N

dimensional space with the alphabets from Σ on each of the N axes.

Formula (3) establishes the relationships among antigen detection rate P, the number of required antibodies in the shape space B, and the complexity of antibodies/antigens N.

The original authors didn't consider antigens as the residents of a shape space. They only analysed selves and antibodies. In theory, it makes sense to partition the shape space Ω into selves and nonselves (antibodies or detectors). So long $\hat{\varepsilon}$ is small enough, the holes caused by selves are negligible, and the shape space is almost fully occupied by antibodies.

However, in practice, we have to study the shape space occupancy by the antigens, because an antigen is put into the shape space to be measured for its distances to the antibodies. The purpose of antibodies in the shape space is to detect antigens. So, we expect, plausibly, that the antibodies can fully cover the whole shape space. Whenever we put an antigen into the shape space, it will be detected by close-by antibodies. This argument implies that the antigens are distributed in the whole shape space, just as the antibodies do. However, real life data sets do not support the implication.

From the point of the view of antigen occupancy in the shape space, there might be a significant portion of the shape space which is unoccupied. These unoccupied locations are caused by arbitrary antibody/antigen representing formats.

For example, in [17], an antibody/antigen is represented by a 50 bits string format. With this format, bits 21-23 (3 bits) can only take the forms of 100, 010, and 001, representing TCP, UDP, and ICMP protocols respectively. The other combinations (5 out of 8) of the 3 bits are impossible. It means that $\frac{5}{8} = 62.5\%$ portion of the shape space cannot be legitimately occupied. Similarly, bits 24-30 (7 bits) only represent 70 different values, while 7 bits can actually represent 128 different values. It then leaves $\frac{128-70}{128} = 45.3\%$ further portion of the shape space unoccupied. Put these 2 segments together, they collectively leave the shape space with 79.5% unoccupied portion. The other bits will further contribute to the unoccupied portion.

In [19], an antibody/antigen is represented by a 49 bits string. The last 8 bits are for the services, a total of 69. Just from the last 8 bits, we can calculate that there will be $\frac{256-69}{256} = 73\%$ portion of the shape space unoccupied.

Similar observations can be made on the other type of data presentation formats, including real number vector formats.

The purpose of generating antibodies is to detect antigens. So, we cannot ignore the landscape of antigen occupancy in the shape space. Now, we know that antigens may not occupy the whole shape space. We have to adapt Formula (3) to this new finding, which actually has significant impact on antibody generation strategy.

Let K be the antigen occupancy rate on the shape space. If we only generate antibodies in the antigen occupied region, Formula (1) becomes

$$\rho = \frac{B}{c_N R^N \times K} \tag{4}$$

and Formula (3) then becomes

$$P = 1 - P_0 = 1 - e^{-\rho c_N \varepsilon^N} = 1 - e^{-\frac{B}{K}\hat{\varepsilon}^N}, \text{ where } \hat{\varepsilon} = \frac{\varepsilon}{R} \tag{5}$$

Formula (5) gives the relationships, under the occupancy rate K, among antigen detection rate P, the number of required antibodies in the shape space B, and the complexity of antibodies/antigens N.

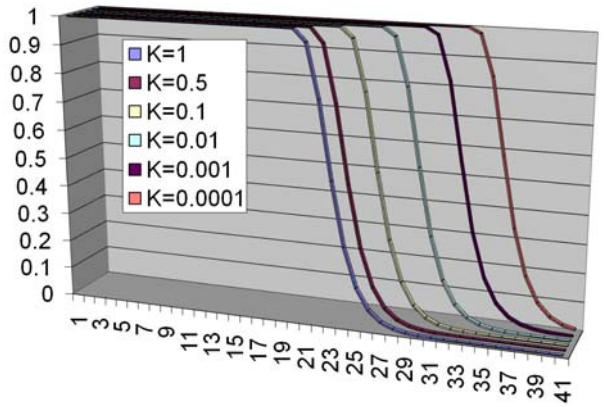


Fig. 1 P (y=P) vs N (x=4N), under shape space occupancy K. Antibodies are distributed only in the antigen occupied region of the shape space.

Fig. 1 shows the detection rate P declines when the complexity N goes up. However, when the generated antibodies only cover the antigen occupied portion of the shape space with respect to the occupancy rate K, as described in Formula (5), the smaller the occupancy rate is, the later the decline begins, which means a larger acceptable N. In Fig. 1, we keep $B = 1 \times 10^7$ and $\hat{\varepsilon} = 0.05$, as they are in [16].

Fig. 1 basically suggests that when generating antibodies, it is essential to take the shape space occupancy rate K into consideration so that the system can cope with large complexity N, which is common with real life application datasets.

III. An Shape Space Occupancy Example

Let's take KDD CUP 1999 dataset as an example. Every vector of the dataset is converted into a 50 bits string [17]. Therefore, the size of the shape space is 2^{50}, which is about 10^{15}. We use the file "kddcup.data" to generate the self strings. We pick up all the vectors with the label "normal" and then obtain 972,781 vectors. After being converted into

bit strings, we obtain 23,587 unique strings. We use the file "corrected" to generate the test strings. The file contains 311,029 vectors, among which 60,593 are labelled as "normal", and the rest 250,436 are labelled with verities of attacks. From the 250,436 vectors with attack labels, we only obtain 12,351 unique strings. Therefore, all together, the occupancy of the shape space by both self and non-self points is:

$$\frac{23587+12351}{10^{15}} = \frac{35938}{10^{15}} \approx 4\times10^{-11} = 4\times10^{-9}\% \quad (8)$$

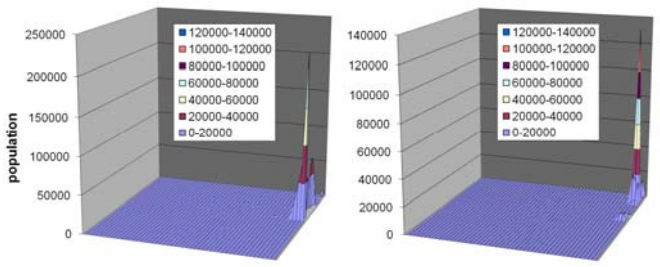
To visually illustrate the shape space occupancy, we project $\{0,1\}^{50}$ space into a 50×50 2-dimensional space. The value 50×50 is chosen for the purpose of visual clarity. For a 50 bits string:

$$b_1b_2...b_{25}b_{26}...b_{49}b_{50} \quad (9)$$

We spilt it into 2 parts: the first 25 bits and the second 25 bits. We use the 2 parts to calculate the coordinate on a 2 dimensional space, the first 25 bits for x and the second 25 bits for y.

$$x = \frac{\log(B2D(b_1b_2...b_{25}))}{\log(2^{25})}\times50, \qquad y = \frac{\log(B2D(b_{26}...b_{49}b_{50}))}{\log(2^{25})}\times50 \quad (10)$$

where $B2D(b_1b_2...b_{25})$ function converts the binary string $b_1b_2...b_{25}$ into its decimal value. To present the repeated points on the shape space, we introduce the concept of the population of a point. For example, for a point, which occupies the shape space at (x,y), the same point may repeatedly appear in the dataset, and they all occupy the exactly same location (x,y) on the shape space. We employ z axis to show the populations of the points on the shape space. The value on the z axis, with respect to (x,y), represents the population of the point on the shape space. Fig. 2(a) illustrates the shape space occupancy by the self points, and Fig. 2(b) is for non-self points.



(a) (b)

Fig. 2 Shape Space Occupancy and the Populations of the Occupying Points

The formulas we used to calculate the (x,y) coordinates are just for shape space occupancy illustration purpose and do not accurately preserve the distances among the points in the original shape space. However, regardless how to calculate, the distance between the 2 exactly same points will always be 0. Therefore, the populations of the points are largely preserved, except for round up errors.

From Fig. 2, we can see that not only is the shape space extremely sparsely occupied, the antigen occupants are clustered closely together. A large number of antigens occupy an exceptionally small portion of the shape space with a large population on each of the occupying locations. *This finding has great impact on antibody generation strategy*.

Finally, although the vectors from KDD CUP 1999 dataset are just some samples of certain time duration and are not exclusive, they do offer a glimpse of the landscape of the antigen occupancy in the shape space for the period of time when the data were collected. We do not expect big deviation from this landscape if more data had been collected. However, even if the complete dataset, should it exist, changes the landscape of the occupancy, the discussion in this section still holds. As discussed in Section II, 79.5% of the shape space is inhabitable due to the particular data presentation format. For the left 20% or so shape space, it is unreasonable to assume that data points will be evenly distributed in the whole habitable region at any given time duration. On the other hand, negative selection has a very strong temporal nature. The newly generated antibodies go through a maturing step. The survivors wait to be activated by the incoming antigens after certain number of matches. The antibodies which do not have enough matches, within a predefined time window, against antigens age and die. Therefore, what matters in negative selection is the dynamics of the involved parties in a time window, not the whole time duration, which may up to eternity.

IV. GENERATING DETECTORS

As demonstrated in the previous sections, a shape space may not be fully occupied. The occupancy rate K could be very, or even extremely, small. In other words, the shape space may be sparsely, perhaps even exceptionally sparsely, occupied. With a very small occupancy rate K, the volume of the unoccupied region of the shape space is significantly big. If ignoring the antigen shape space occupancy, the unoccupied volume will require a large number of detectors to cover, whereas any detector in this region is effectively useless, even worse as it also incurs overhead on computational resources and performance, because the region cannot possibility be inhabited by any antigen. Therefore, in practice, *it is neither necessary nor feasible to generate detectors to cover the whole shape space.*

If we only generate the detectors which just cover only the habitat region of the shape space, the number of the required detectors is significantly smaller. We call this type of detectors *effective detectors*. As a consequence, the performance of negative selection algorithms will be significant improved. Thus, the scalability problem is

solved, or at least, greatly alleviated.

With real life datasets, as seen from the example in Section III, data items hardly follow the ideal Poisson distribution, especially for abnormal network traffic and spam email etc., where data are highly irregular with burst nature, e.g., "*power-law distribution*" and "*one-sided and heavy tailed*" [20]. From the example, we can conclude that in some applications, the occupants in the habitat region of the shape space tend to cluster together. Therefore, it is prudent to generate detectors around the occupants. This heuristic technique may result even less number of required detectors (hence, better performance), yet better detection rate. It further alleviates the scalability problem.

```
self set S, detector set D, and antigen set G

Procedure GenerateADetector (seed)
{
  if (seed == nil) {
    generate a random detector candidate c
  }
  else { c = seed; }

  foreach (s in S) {
    if (c matches s) eliminating c;
  }
  if (c != nil) {
    Mature(c); // eliminating c if it can't be activated
    if (c != nil) { add c to D}
  }
}

Procedure Detecting()
{
  foreach (g in G) {
    foreach (d in D) {
      if (g matches d) { g is non-self; break; }
    }
    if (g does not match any d in D) {
      g is regarded self (but possibly wrong);
      GenerateADetector (g);
    }
  }
}
```

Fig. 3 The antigen feedback mechanism in a negative selection algorithm (Only the concerned components are listed, and concurrent execution is ignored.)

In summary, the heuristics for generating detectors are:

- generate only the effective detectors which cover the habitat region of a shape space, and
- generate the detectors which are around the occupants in the habitat region.

An antigen feedback mechanism, Fig. 3, initially proposed in [17], fulfils both requirements. With the antigen feedback mechanism, an unmatched antigen is treated as a newly generated detector. It goes through the same maturing process and is subject to elimination if it matches any of the self strings. If it survives, it will be used to match further incoming antigens. If it can be activated by exceeding the pre-defined activation threshold, it becomes a legitimate detector. The detectors generated by this method are around the occupants in the habitat region. Therefore, they are not far away from the occupants in the habitat region. It is thus rare for them to fall into the unoccupied region of the shape space.

There are 2 different approaches in selection orientation for anomaly detection purpose, positive selection and negative selection [27]. A system may keep the samples of selves. Any match against the self samples means the detection of a harmless self, while no match means a harmful antigen. The system is therefore of positive selection. On the other hand, a system can keep non-self detectors, where any match means the detection of a harmful antigen, while no match means a harmless self. This system is of negative selection.

Negative selection with the antigen feedback mechanism has 2 advantages over positive selection. First, the number of detectors is much less than the number of selves in the system, therefore, much better detection performance. As we discussed previously, a negative selection system has strong temporal nature. What matters in a negative selection system is the dynamics of the involved parties in a fixed time window, not the whole time duration. Consequently, only these active detectors of the current time window are needed in a negative selection system. In a positive selection system, the samples of selves have to cover the while duration of the execution, not just for a fixed time window. This observation has been verified repeatedly in our experiments. Second, in a positive selection system, all detected harmful antigens are regarded the same, belonging to the single harmful antigen class. While in a negative selection system, by querying the attributes of the detectors, we can learn more information, such as patterns and frequencies etc., about the detected harmful antigens. This potential of negative selection makes it an excellent candidate for pattern discovery, which might be an area very well just for negative selection systems. We are currently investigate this potential application of negative selection [28].

V. Experiment Results and Discussions

In this section, we report the results of our experiments on 4 different datasets, with focusing on the impact of antigen feedback mechanism on effective detectors. The 4 experiments cover the data presentation formats of both strings and real number valued vectors.

Our first set of experiments was on KDD CUP 1999 dataset. Each of the original data item was converted into a string of 50 bits [17]. In the experiment, we used r-continuous bits match rule [1, 2] to calculate the distance between a detector and an antigen. Without the antigen feedback mechanism, just by randomly generating detectors to cover the whole shape space, we failed to generate a single useful detector in the duration of 3 continuous days. This is understandable, because we were experimenting on $4\times10^{-9}\%$ odd. With the antigen feedback mechanism, any single particular experiment can be completed within a few

hours. When r=33, with the activation matching threshold set at 5, we achieved the detection rate 98.9% for normal

TABLE I
THE NUMBER OF DETECTORS AND THE DETECTION RATES

Dataset	No. of atg's	No. of detectors	Detection rate
KDD Cup 99	20000	All: 35	Normal: 98.9%
		90% detection: 21	Intrusion: 95.19%
TREC07	20000	All: 876	Ham: 94%
		90% detection: 260	Spam: 78%
Iris	150	All: 2	Setosa: 100%
		90% detection: 1	Non Setosa: 97%
Triangle	1000	All: 21	Normal: 100%
		90% detection: 14	Abnormal: 86%

strings and 95.19% for attacking strings, respectively. While the number of all detectors used over the whole experiment duration is 35. Among the 35 detectors, 21 are responsible for 90% of the detection. The number of selves is 558.

Our experiments on the second dataset were performed on TREC07 spam email dataset [23]. We first converted the body part (only the body part) of each of the emails into its Nilsimsa digest [26], which is a 256-bits string. If the total number of the difference of the corresponding bits between 2 digests exceeds a threshold (the predefined affinity threshold), the 2 original emails which produce the 2 digests are regarded as unrelated, i.e., no match between the two. TREC07 corpus has 75,419 emails. An experiment on all the emails takes a quite long time to complete, while the experiment results on all the emails are actually similar to these on the first 20,000 emails. The only difference is that for a full run with all 75,419 emails, the best detection rates, with respect to both ham and spam, were achieved when the affinity threshold was set at around 90, instead of at around 80. In this paper, we only report our experiment results for the first 20,000 emails. Of the 20,000 emails, there are 4,890 ham emails and 14,489 spam emails. There are also 621 blanks (620 from spam emails and 1 from ham email), which were skipped in the experiments. The 20,000 emails only yield 12,864 unique Nilsimsa digests, which suggests the occupancy of the shape space during this observing time window is at $12864/2^{265} \approx 12864/10^{80} \approx 1.3\times10^{-76}$.

We used the first 2500 ham emails (about 50% of the ham emails) as the seeds for selves. Therefore, the number of selves is 2500. We focused on using negative selection in detecting new strains of spam emails. Therefore, we didn't use any knowledge of known spam emails, because, under the assumption, they were supposed to be new strains, and no knowledge about them was supposed to be available yet. Except for 2,500 seeding ham emails, the system does not have any information about any spam email, and it has to learn the spam email patterns on the fly by itself. The best outcome achieved 94% detection rate on ham email and 78% detection rate on spam email when the affinity threshold was set at 80 and the detector activation threshold was set at 3 [22]. Although the results are not as good in comparison with other approaches, they are justifiable under the assumption – no available prior knowledge about spam emails. Of course, with the help of all available information, say, email header lines, suspicious keywords, and heuristic rules based on the previous knowledge of spam email etc., a system can considerably increase the accuracy of spam email detection, especially when on fixed spam email corpora. In the round of our experiment which produced the best outcomes, 876 detectors were generated through the antigen feedback mechanism. Among them, 260 were responsible for 90% of the detection.

The third dataset used in our experiments is the famous Iris dataset [24]. We choose the 50 Setosa data items as normal, and the rest as abnormal. So, the number of selves is 50. After being given the 50 Setosa data items (real number valued vectors) as selves, the system learns the patterns of non-Setosa data items by itself. It achieved the detection rate of 100% for Setosa and 97% for non-Setosa respectively. Surprisingly, only 2 detectors were needed, with 1 of them detected 99% of the non-Setosa data items, and other one only 1%. We used Euclidean distance in these experiments to calculate the affinity between 2 data items. The threshold, under which 2 data items were considered the same, was set at between 2.7 and 3.5, and the antibody activation threshold was set at 3 matches.

The last dataset for our experiments is the 2D synthetic dataset created by Dipankar Dasgupta. We obtained the dataset from Keogh [25]. The data items in this dataset are almost evenly distributed on a limited 2D space. However, if we exam the 2D space carefully, the occupied area is far less than the unoccupied one. A point, which corresponds to a data item in the dataset, occupies the round area with the center of the coordination from the data item and the radius of the affinity threshold. With Triangle-small dataset, if we set the affinity threshold at 0.16, and the antibody activation threshold at 2 matches, we can achieve the detection rate of 100% for normal data items and 86% for abnormal data items. The number of the detectors in use was 21. Among them, 14 detectors performed 90% of the detection. The number of selves is 1000.

Table 1 lists the numbers of detectors and detection rates of the 4 sets of the experiments. We'd like to emphasize again that every detection rate for abnormal data listed in the table bears the failures of the initial learning phase of the corresponding experiment. In other words, no information about the abnormal data was given to any of the experiments, and the abnormal data patterns were discovered on the fly during the experiment.

VI. SUMMARY AND FUTURE WORK

In this paper, we first briefly review the original shape space theory. Realizing that a shape space may not be fully occupied due to antibody/antigen data presentation formats, we introduce the concept of shape space occupancy rate. We then further study the impact of the space occupancy rate on the relationships among antigen detection rate, the number

of required antibodies in the shape space, the complexity of antibodies/antigens, and the space occupancy rate. Based on the outcomes of the study, we propose the heuristics for generating detectors to solve the scalability problem of negative selection. We also discuss the 2 advantages a negative selection system has over a positive selection system. Finally, we report our experiments on 4 different datasets, with focusing on the performance of antigen feedback mechanism on the numbers of effective detectors and detection rates.

In the near future, we will conduct more experiments on a broad range of datasets. We anticipate that the experiments will further confirm the validity of our studies in this paper in a broad range of applications, and our proposal of the antigen feedback mechanism will result efficient negative selection processes.

ACKNOWLEDGMENT

The authors sincerely thank the anonymous reviewers for their thorough feedback, which greatly improves the quality of this paper.

REFERENCES

[1] Hofmeyr, S.A. and S. Forrest. Immunity by Design: An Artificial Immune System. in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999). 1999. Orlando, Florida, USA: Morgan Kaufmann.

[2] Hofmeyr, S.A. and S. Forrest, Architecture for an Artificial Immune System. Evolutionary Computation, 2000. 8(4): p. 443-473.

[3] Timmis, J., Artificial immune systems - today and tomorrow. Natural Computing: an international journal, 2007. 6(1): p. 1-18.

[4] Dasgupta, D., Advances in artificial immune systems. IEEE Computational Intelligence Magazine, 2006. 1(4): p. 40 - 49.

[5] Hart, E. and J. Timmis. Application Areas of AIS: The Past, The Present and The Future. in Proceedings of Artificial Immune Systems: 4th International Conference, ICARIS 2005. 2005. Banff, Alberta, Canada: Springer.

[6] Castro, L.N.D. and J. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach. 2002: Springer.

[7] Balthrop, J., S. Forrest, and M.R. Glickman. Revisiting LISYS: Parameters and normal behavior. in Proceedings of the Congress on Evolutionary Computing (CEC-2002). 2002.

[8] Gabrielli, N. and M. Rigodanzo. An Artificial Immune System for Network Intrusion. Detection on a Web Server: First Results. in Proceedings of the 2nd Italian Workshop on Evolutionary Computation (GSICE 2006). 2006.

[9] Gonzalez, F.A. and D. Dasgupta, Anomaly Detection Using Real-Valued Negative Selection. Genetic Programming and Evolvable Machines, 2003. 4(4): p. 383-403.

[10] Dasgupta, D., K. Kumar, D. Wong, and M. Berry. Negative Selection Algorithm for Aircraft Fault Detection. in Proceedings of the Third International Conference on Artificial Immune Systems (ICARIS 2004). 2004.

[11] Ji, Z. and D. Dasgupta, Revisiting Negative Selection Algorithms. Evolutionary Computation, 2007. 15(2): p. 223-251.

[12] Kim, J. and P. Bentley. An evaluation of negative selection in an artificial immune system for network intrusion detection. in Proceedings of GECCO-2001. 2001.

[13] Stibor, T. An Empirical Study of Self/Non-self Discrimination in Binary Data with a Kernel Estimator. in 7th International Conference on Artificial Immune Systems (ICARIS 2008). 2008.

[14] Stibor, T., P. Mohr, J. Timmis, and C. Eckert. Is negative selection appropriate for anomaly detection? in Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO'05). 2005.

[15] Stibor, T., K.M. Bayarou, and C. Eckert. An Investigation of R-Chunk Detector Generation on Higher Alphabets. in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004). 2004. Seattle, WA, USA: Springer.

[16] Perelson, A.S. and G.F. Oster, Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-non-self discrimination. Journal of theoretical biology, 1979. 81(4): p. 645-670.

[17] Ma, W., D. Tran, and D. Sharma. Negative Selection with Antigen Feedback in Intrusion Detection. in 7th International Conference on Artificial Immune Systems (ICARIS 2008). 2008.

[18] ACM. KDD CUP 1999 data. [cited 12 January 2007]; Available from: http://kdd.ics.uci.edu//databases/kddcup99/kddcup99.html.

[19] Hofmeyr, S., An Immunology Model of Distributed Detection and Its Application to Computer Security, in Department of Computer Science. 1999, University of New Mexico, USA.

[20] Chan, P.K., M.V. Mahoney, and M.H. Arshad, A Machine Learning Approach to Anomaly Detection. 2003, Florida Institute of Technology: Melbourne, FL, USA.

[21] Elberfeld, M. and Textor, J. Efficient Algorithms for String-Based Negative Selection, in 8th International Conference on Artificial Immune Systems (ICARIS 2009). 2009.

[22] Ma, W., D. Tran, and D. Sharma. A Novel Spam Email Detection System Based on Negative Selection, in 4th ICCIT: 2009 International Conference on Computer Sciences and Convergence Information Technology. 2009.

[23] Cormack, G. and T. Lynam. 2007 TREC Public Spam Corpus, http://plg.uwaterloo.ca/~gvcormac/treccorpus07/. 2007 [cited 15 Janurary 2009].

[24] Asuncion, A. & Newman, D.J. (2007). UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/Iris. Irvine, CA: University of California, School of Information and Computer Science

[25] Keogh, E, General Time Series Tutorial. [cited 20 December 2009]; Available from: http://www.cs.ucr.edu/~eamonn/Keogh_Time_Series_CDrom.zip.

[26] Prakash, V.V. Digest-Nilsimsa, http://search.cpan.org/dist/Digest-Nilsimsa/. 2002 [cited 20 February 2009].

[27] Stibor, T., Mohr, P.,Timmis, J. and Eckert, C. Is negative selection appropriate for anomaly detection? GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation. pp 321-328.

[28] Ma, W., D. Tran, and D. Sharma. Negative Selection as a Means of Discovering Unknown Temporal Patterns. To be published in ICIS 2010, International Conference on Intelligent Systems.